**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 22 June 3035 |
| --- | --- |
| Team ID | LTVIP2025TMID54182 |
| Project Name | LearnHub: Your Center for Skill Enhancement |
| Maximum Marks | 4 Marks |

### ****Technical Architecture****

This Online Learning Platform (OLP) follows a **Client-Server Architecture** using the MERN Stack:

* **Frontend (Client):** React.js (with Vite, Bootstrap, Material UI)
* **Backend (Server):** Node.js + Express.js
* **Database:** MongoDB (NoSQL)
* **Hosting (Optional):** Can be deployed using Render, Vercel, Netlify, or cloud providers like AWS/GCP.

**Infrastructural Demarcation:**

* Frontend and backend can be hosted independently.
* MongoDB can be cloud-hosted via MongoDB Atlas.





### Table-1: Components & Technologies

| **S.No** | **Component** | **Description** | **Technology/Tool Used** |
| --- | --- | --- | --- |
| 1 | User Interface | Web UI for students, teachers, admin | React.js, Vite, HTML, CSS, Bootstrap, Material UI |
| 2 | Application Logic-1 | User Authentication and Authorization (JWT) | Node.js, Express.js, JWT |
| 3 | Application Logic-2 | Course Management, Enrollment, Role-based Access | Express.js, Node.js, REST APIs |
| 4 | Application Logic-3 | Payment & Certificate Generation | Stripe API (optional), HTML-PDF |
| 5 | Database | Stores users, courses, enrollments | MongoDB, Mongoose ODM |
| 6 | Cloud Database | Cloud-hosted version of MongoDB (if deployed) | MongoDB Atlas |
| 7 | File Storage | Store course images/videos | Multer (Local Storage) or Cloudinary/AWS S3 |
| 8 | External API-1 | Payment Integration | Stripe/PayPal API |
| 9 | External API-2 | Email Notifications | Nodemailer or SendGrid |
| 10 | Machine Learning Model | Not Applicable (NA) | - |
| 11 | Infrastructure | Hosting for frontend/backend | Localhost, Vercel, Render, or AWS |

### Table-2: Application Characteristics

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Frameworks used in frontend/backend | React.js, Express.js, Node.js, Mongoose |
| 2 | Security Implementations | JWT Auth, Role-Based Access, bcrypt password hashing | JWT, bcrypt.js, dotenv |
| 3 | Scalable Architecture | Separated frontend/backend; scalable via cloud hosting | MERN Stack, RESTful API |
| 4 | Availability | Can be distributed via multi-server/cloud setup | Load Balancer (Nginx)/Cloud-based Hosting |
| 5 | Performance | Optimized with React, REST APIs, Vite build system | Vite, Axios, Caching if needed |

### ****References****

* <https://c4model.com/>
* <https://developer.ibm.com/patterns/>
* <https://www.mongodb.com/atlas>
* <https://vitejs.dev/>
* <https://expressjs.com/>
* <https://react.dev/>